Πληροφοριακά Συστήματα

Διάλεξη 4 (2 Απρ. 2024)

Διονύσης Μάργαρης
Επίκουρος Καθηγητής ΤΨΣ ΠΑΠΕΛ

Τι θα συζητήσουμε σήμερα;

Πλεονασμός Πληροφορίας

Πλεονασμός Πληροφορίας (1/2)

Τα λάθη στα δεδομένα συμβαίνουν όταν εκείνα μεταφέρονται από τη μια μονάδα στην άλλη, από ένα σύστημα σε ένα άλλο, ή όταν αυτά αποθηκεύονται σε μια μονάδα μνήμης. Για να ανεχτούμε τέτοια λάθη, εισάγουμε την έννοια του πλεονασμού στα δεδομένα, που καλείται πλεονασμός πληροφορίας (information redundancy).

Η πιο κοινή μορφή πλεονασμού πληροφορίας είναι η κωδικοποίηση (coding), που προσθέτει bits στα δεδομένα, επιτρέποντάς μας να πιστοποιήσουμε την ορθότητα των δεδομένων πριν τα χρησιμοποιήσουμε, και σε κάποιες περιπτώσεις, μπορούμε ακόμη και να διορθώσουμε τα εσφαλμένα bits δεδομένων.

Πλεονασμός Πληροφορίας (2/2)

Η εισαγωγή του πλεονασμού πληροφορίας μέσω κωδικοποίησης δεν περιορίζεται στο επίπεδο μεμονωμένων λέξεων δεδομένων αλλά μπορεί να επεκταθεί για να παρέχει ανθεκτικότητα σε σφάλματα για μεγαλύτερες δομές δεδομένων.

Το πιο δημοφιλές παράδειγμα τέτοιας χρήσης είναι
το σύστημα αποθήκευσης Πλεονάζοντος Πίνακα Ανεξάρτητων Δίσκων (Redundant Array of Independent Disks – RAID).

Κωδικοποίηση (1/2)

Όταν κωδικοποιούμε, μια λέξη δεδομένων d-bits κρυπτογραφείται (encoded) σε μια κωδική λέξη (codeword) c-bits, με c > d.
Αυτή η κρυπτογράφηση εισάγει τον πλεονασμό πληροφορίας, κάτι που μας ωθεί να χρησιμοποιήσουμε πιο πολλά bits από όσα χρειαζόμαστε.

Μια συνέπεια αυτού του πλεονασμού πληροφορίας είναι ότι δεν αποτελούν όλοι οι 2^c δυαδικοί συνδυασμοί των c bits έγκυρες κωδικές λέξεις.

Κωδικοποίηση (2/2)

Σαν αποτέλεσμα, όταν επιχειρούμε να αποκρυπτογραφήσουμε (decode) την λέξη των c bits για να εξάγουμε τα αρχικά d bits δεδομένων, ίσως έρθουμε αντιμέτωποι με μια άκυρη κωδική λέξη, κάτι που σημαίνει πως συνέβη λάθος.

Έχουμε 3 περιπτώσεις κατά τη φάση της αποκωδικοποίησης:

- Δεν αντιλαμβανόμαστε κανένα λάθος (χωρίς να σημαίνει πως δεν υπάρχει)
- Αντιλαμβανόμαστε πως υπάρχει κάποιο λάθος χωρίς να είμαστε σε θέση να πούμε που έγινε το λάθος
- Αντιλαμβανόμαστε το λάθος και είμαστε σε θέση να πούμε που ακριβώς έγινε το λάθος και να το διορθώσουμε

Είδη Κωδικοποίησης (1/2)

- Κώδικες Ισοτιμίας (στέλνουμε 0, αν έχουμε ζυγό πλήθος άσσων, αλλιώς 1)
- Κώδικας Hamming (προσθέτουμε r bits, με $2^r \geq d + r + 1$ ώστε να πετύχουμε ανίχνευση λάθους και στα bits που προσθέτουμε)
- Άθροισμα Ελέγχου (προσθέτουμε τα bytes πληροφορίας και στέλνουμε κ' το άθροισμα – συνηθίζεται για έλεγχο κολλημένης γραμμής στο 0)
- Κώδικες M-of-N (κάθε κωδική λέξη των N bits έχει ακριβώς M bits που είναι μονάδα, οδηγώντας σε κωδικές λέξεις. Οποιοδήποτε λάθους απλού bit θα αλλάξει το πλήθος των μονάδων σε M + 1 είτε σε M – 1 και θα ανιχνευτεί)

Είδη Κωδικοποίησης (2/2)

- Κυκλικοί και Αριθμητικοί Κώδικες (η κρυπτογράφηση των δεδομένων αποτελείται από τον πολλαπλασιασμό (modulo-2) της λέξης δεδομένων με έναν σταθερό αριθμό. Το γινόμενο αποτελεί και την κωδική λέξη. Η αποκρυπτογράφηση γίνεται με τη διαίρεση με την ίδια σταθερά: αν το υπόλοιπο δεν είναι μηδενικό, φαίνεται ότι έχει συμβεί κάποιο λάθος)

- Κώδικας Berger (στέλνουμε μαζί και το συμπλήρωμα του πλήθους των άσσων)

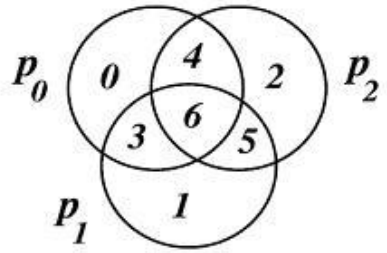
Παράδειγμα - Εφαρμογή Κώδικά Hamming

Έχουμε d = 4 bits δεδομένων, a3a2a1a0, άρα r = 3 είναι ο ελάχιστος αριθμός bits ισοτιμίας (4 + **3** + 1 <= 8 = 2^3), τα p2p1p0.

Η ολοκληρωμένη κωδική λέξη των 7 bits είναι a3a2a1a0p2p1p0.

Η ανάθεση των bits ισοτιμίας των καταστάσεων

Κατάσταση	Εσφαλμένοι έλεγχοι ισοτιμίας	Σύνδρομο
Καθόλου λάθη	Κανένας	000
Bit 0 (p_0) λάθος	p_0	001
Bit 1 (p_1) λάθος	p_1	010
Bit 2 (p_2) λάθος	p_2	100
Bit 3 (a_0) λάθος	p_0, p_1	011
Bit 4 (a_1) λάθος	p_0, p_2	101
Bit 5 (a_2) λάθος	p_1, p_2	110
Bit 6 (a_3) λάθος	p_0, p_1, p_2	111



Παράδειγμα - Εφαρμογή Κώδικά Hamming

Το p0 καλύπτει τις θέσεις 0, 3, 4 και 6,

Άρα p0 = α0⊕ α1⊕ α3.

Αντίστοιχα p1 = a0⊕a2⊕ a3

και p2 = a1⊕ a2⊕a3.

Για παράδειγμα, για α3α2α1α0 = 1100, έχουμε p2p1p0 = 001.

Έστω ότι η ολοκληρωμένη κωδική λέξη 1100001 αντιμετωπίζει ένα απλό λάθος bit και γίνεται 1**0**00001.

Υπολογίζουμε ξανά τα τρία bits ισοτιμίας με βάση μόνο τα a3a2a1a0 και παίρνουμε p2p1p0 = 111.

Άρα, λάθος bits ισοτιμίας τα p2 και p1. Άρα το λάθος είναι στο bit5 -> a2.

Εύκαμπτα Συστήματα Δίσκων

Πλεονασμός πληροφορίας μέσω κωδικοποίησης σε υψηλότερο επίπεδο από τις μεμονωμένες λέξεις δεδομένων είναι η δομή RAID (Redundant Arrays of Independent/Inexpensive Disks), ή αλλιώς των Πλεοναζόντων Πινάκων Ανεξάρτητων/Οικονομικών Δίσκων.

Υλοποίηση RAID

Επίπεδο Υλικού

- Συνήθως γίνεται είτε από έναν αφοσιωμένο RAID ελεγκτή, είτε από έναν ελεγκτή SCSI. Οι ελεγκτές SCSI ποικίλουν από την άποψη της ικανότητάς τους να υποστηρίζουν διάφορα επίπεδα RAID.

Επίπεδο Λογισμικού

- Μπορεί να οριστεί και σε επίπεδο λογισμικού από το λειτουργικό σύστημα.

  Ένα λειτουργικό σύστημα που υποστηρίζει RAID εσωτερικά είναι τα Windows NT.

Γενικά είναι πιο συνηθισμένο το RAID σε επίπεδο H/W επειδή είναι πιο γρήγορο.

Επιλογές RAID

- ✓ Standard RAID levels
- ✓ Nested RAID levels
- ✓ Non-standard RAID levels

Standard RAID levels

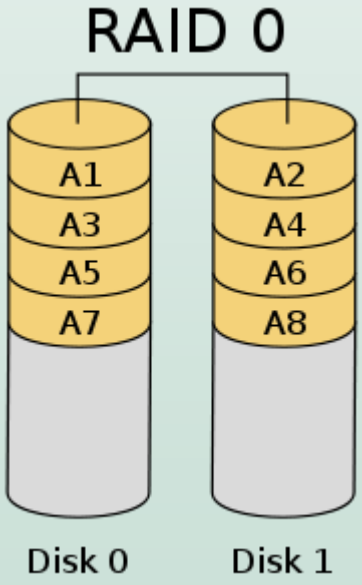
- RAID 0 (or just AID?)
- RAID 1
- RAID 2
- RAID 3
- RAID 4
- RAID 5
- RAID 6

RAID 0 (stripe set or striped volume)

Στο RAID 0 τα δεδομένα χωρίζονται ομοιόμορφα σε δύο ή περισσότερους δίσκους (λωρίδες – strips) χωρίς καμία ισοτιμία πληροφορίας για πλεονασμό.

Δεν υπάρχει κανένας πλεονασμός πληροφορίας και χρησιμοποιείται μόνο για την αύξηση της απόδοσης

Ένα RAID 0 μπορεί να δημιουργηθεί με δίσκους διαφορετικού μεγέθους, αλλά το μέγεθος του κάθε δίσκου που θα είναι ωφέλιμο θα είναι ίσο με το μέγεθος του μικρότερου που συμμετέχει στο RAID.



RAID 0 παράδειγμα

Ας υποθέσουμε ότι έχουμε 2 σκληρούς δίσκους μεγέθους 120GB και 100GB. Αν θέλουμε να υλοποιήσουμε ένα RAID 0 σύστημα, τότε αυτό θα έχει μέγεθος = 2*min(120GB,100GB) = 2*100GB = 200GB.

Το RAID 0 μπορεί να υλοποιηθεί και για περισσότερους από 2 δίσκους, στις περισσότερες υλοποιήσεις όμως συμμετέχουν 2 δίσκοι, αφού το σύστημα αποτυγχάνει ακόμα και αν 1 μόνο δίσκος αποτύχει, άρα η αξιοπιστία μειώνεται όσο περισσότερους δίσκους έχουμε.

Αν, για παράδειγμα, υπάρχει 5% πιθανότητα κάποιος δίσκος να χαλάσει μέσα σε 3 χρόνια, τότε η πιθανότητα να αποτύχει ένα RAID 0 σύστημα 2 δίσκων είναι ίσο με:

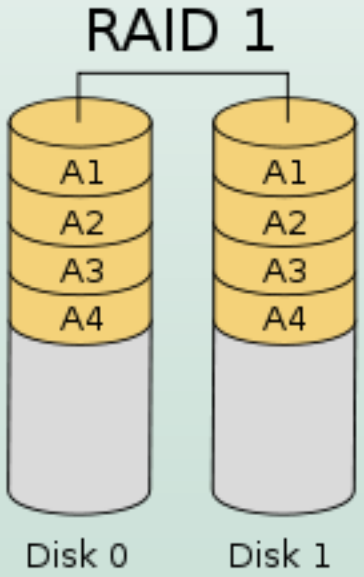
$$\mathbb{P}(\text{at least one fails}) = 1 - \mathbb{P}(\text{neither fails}) = 1 - (1 - 0.05)^2 = 0.0975 = 9.75\,\%$$

RAID 1 (mirrored disks) (1/3)

Το επίπεδο 1 των RAID (RAID1) αποτελείται από είδωλα δίσκων (mirrored disks). Στη θέση ενός δίσκου υπάρχουν δύο, με τον καθένα να είναι ένα αντίγραφο του άλλου.

Αν ένας δίσκος πάθει βλάβη, ο άλλος μπορεί να συνεχίσει να εξυπηρετεί αιτήσεις πρόσβασης. Αν λειτουργούν ορθά και οι δύο δίσκοι, το RAID1 μπορεί να επιταχύνει την ανάγνωση των προσβάσεων μοιράζοντάς τις στους δύο δίσκους.

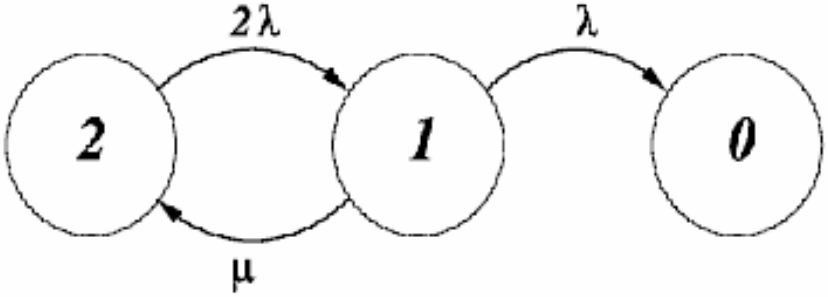
Η εγγραφή των προσβάσεων, ωστόσο, καθυστερεί επειδή οι δύο δίσκοι πρέπει να τελειώσουν πρώτα την ενημέρωση (update) πριν την περάτωση της λειτουργίας.



RAID 1 (2/3)

Ας υποθέσουμε ότι οι δίσκοι παθαίνουν βλάβες ανεξάρτητα ο ένας από τον άλλον, ο καθένας με έναν σταθερό ρυθμό λ, και ότι ο χρόνος επιδιόρθωσης του καθένα είναι εκθετικά κατανεμημένος με παράμετρο μ.

Για να υπολογίσουμε την αξιοπιστία, θεωρούμε μια αλυσίδα Markov τριών επιπέδων.



Αλυσίδα Markov για υπολογισμό αξιοπιστίας RAID1 .

RAID 1 (3/3)

Η μη-αξιοπιστία σε χρόνο t είναι η πιθανότητα το σύστημα να βρίσκεται σε κατάσταση βλάβης, $P_0(t)$. Οι διαφορικές εξισώσεις που σχετίζονται με αυτήν την αλυσίδα Markov είναι οι ακόλουθες:

$$\frac{dP_2(t)}{dt} = -2\lambda P_2(t) + \mu P_1(t)$$

$$\frac{dP_1(t)}{dt} = -(\lambda+\mu)P_1(t) + 2\lambda P_2(t)$$

$$P_0(t) = 1 - P_1(t) - P_2(t)$$

$$A = \frac{\mu(\mu+2\lambda)}{(\lambda+\mu)^2}$$

$$MTTDL = \sum_{n=1}^{\infty} q^{n-1} p T_{2\to 0}(n) = \sum_{n=1}^{\infty} n q^{n-1} p T_{2\to 0}(1) = T_{2\to 0}(1)/p = \frac{3\lambda+\mu}{2\lambda^2}$$

$$R(t) \approx e^{-t/MTTDL}$$

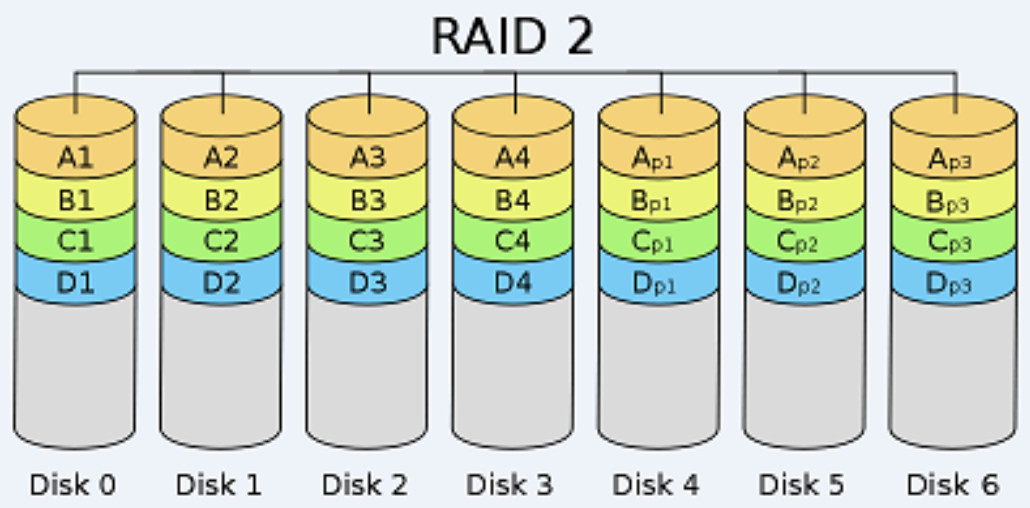
RAID 2 (1/2)

Το RAID Επιπέδου 2 αποτελείται από μια τράπεζα δίσκων δεδομένων σε παράλληλους δίσκους κωδικοποιημένους με Hamming.

Υποθέστε ότι υπάρχουν d δίσκοι δεδομένων και c δίσκοι κωδίκων.
Τότε, μπορούμε να θεωρήσουμε το i-οστό bit του κάθε δίσκου σαν bits μιας λέξης (c + d) bits. Βάσει της θεωρίας των κωδίκων Hamming, γνωρίζουμε πως πρέπει να έχουμε $2^c \geq c + d + 1$ για να είμαστε ικανοί να διορθώσουμε το ένα bit ανά λέξη.

RAID 2 (2/2)

Για παράδειγμα, η χρήστη του κωδικού Hamming(7,4), που σημαίνει 4 bits δεδομένων συν 3 bits για διόρθωση λαθών, ισοδυναμεί με τη χρήση 7 σκληρών δίσκων, εκ των οποίων οι 4 χρησιμοποιούνται για αποθήκευση δεδομένων και οι 3 για διόρθωση λαθών.

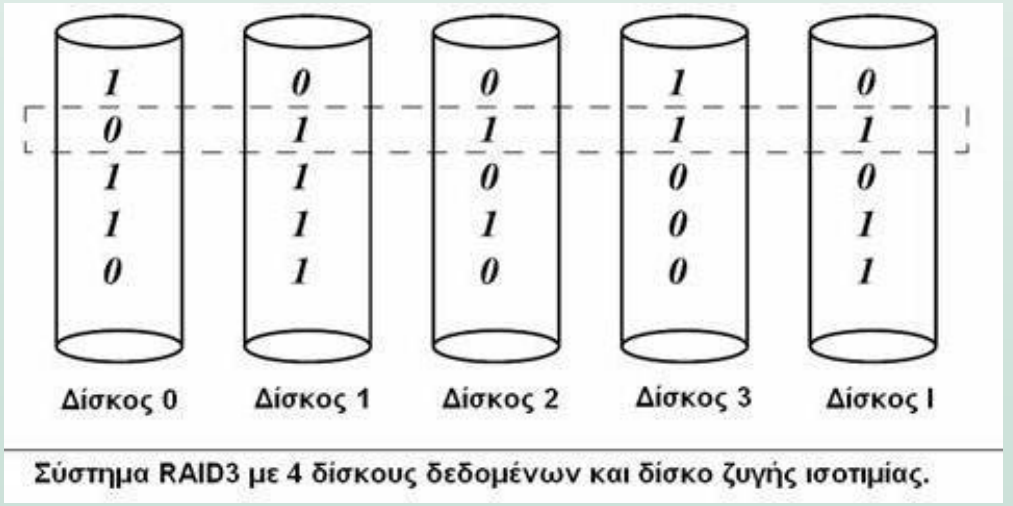


Λόγω του φόρτου που επιφέρει αυτό το επίπεδο RAID (σε σχέση με τα υπόλοιπα) χρησιμοποιείται πολύ σπάνια.

RAID 3 (1/2)

Το RAID3 αποτελεί μια τροποποίηση του RAID2 και προκύπτει από την παρατήρηση ότι κάθε δίσκος έχει κωδικοποίηση διόρθωσης λάθους ανά τομέα. Επομένως, αν κάποιος τομέας πάθει βλάβη, μπορούμε να τον αναγνωρίσουμε.

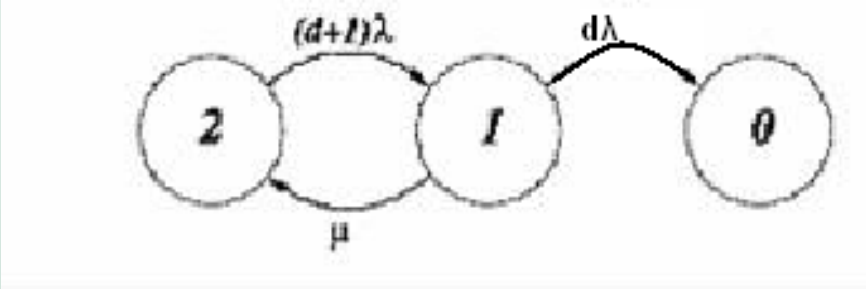
Το RAID3 αποτελείται από μια τράπεζα d δίσκων δεδομένων μαζί με έναν δίσκο ισοτιμίας. Τα δεδομένα είναι διαστρωμένα με bits (bit-interleaved) κατά μήκος των δίσκων δεδομένων και η i-οστή θέση bit της ισοτιμίας δίσκου περιέχει το bit ισοτιμίας που σχετίζεται με τα bits στην i-οστή θέση του κάθε δίσκου δεδομένων.



Σύστημα RAID3 με 4 δίσκους δεδομένων και δίσκο ζυγής ισοτιμίας.

RAID 3 (2/2)

Οι αλυσίδες Markov για την αξιοπιστία και τη διαθεσιμότητα αυτού του συστήματος είναι σχεδόν ίδιες με εκείνες που χρησιμοποιήθηκαν στο RAID1. Εκεί, είχαμε δύο δίσκους ανά σύνολο. Εδώ, έχουμε d + 1. Και στις δύο περιπτώσεις, το συνολικό σύστημα παθαίνει βλάβη (έχουμε απώλεια δεδομένων) αν δύο η περισσότεροι δίσκοι πάθουν βλάβη.



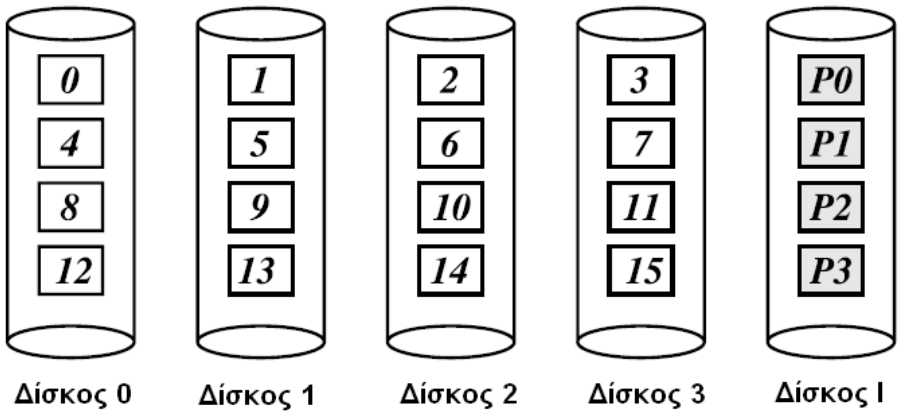
Αλυσίδα Markov για υπολογισμό αξιοπιστίας RAID3.

$$MTTDL = \frac{(2d+1)\lambda + \mu}{d(d+1)\lambda^2}$$

$$R(t) \approx e^{-t/MTTDL}$$

RAID 4 (1/2)

Το RAID4 είναι παρόμοιο με το RAID3, εκτός από το γεγονός ότι η μονάδα που παρεμβάλλεται δεν αποτελείται από ένα bit αλλά από ένα μπλοκ αυθαίρετου μεγέθους, που καλείται ράβδωση (stripe).



Εικόνα 3.20 Σύστημα RAID4 με τέσσερις δίσκους δεδομένων και ένα δίσκο ισοτιμίας (κάθε ορθογώνιο της εικόνας περιέχει ένα μπλοκ δεδομένων).

RAID 4 (2/2)

- Το πλεονέκτημα του RAID4 έναντι του RAID3 είναι ότι μια μικρή λειτουργία ανάγνωσης μπορεί να περιοριστεί μέσα σε έναν μόνο δίσκο δεδομένων και όχι να παρεμβληθεί σε όλους.
Έτσι, οι μικρές αναγνώσεις είναι ταχύτερες στο RAID4 παρά στο RAID3.

- Μια παρόμοια παρατήρηση ισχύει και για τις μικρές εγγραφές: σε μια τέτοια λειτουργία, και ο επηρεαζόμενος δίσκος δεδομένων και ο δίσκος ισοτιμίας πρέπει να αναβαθμιστούν.
Η αναβάθμιση της ισοτιμίας είναι σχετικά απλή: το bit ισοτιμίας εναλλάσσεται (toggles) αν το αντίστοιχο bit δεδομένων που γράφεται είναι διαφορετικό από εκείνο που εκείνο που αντικαθίσταται.

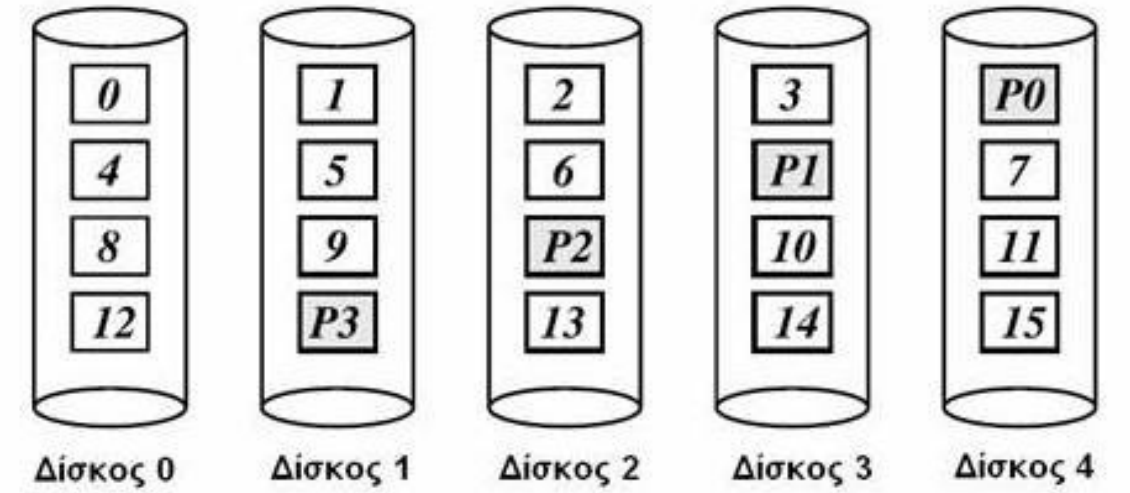
- Το μοντέλο αξιοπιστίας του RAID4 είναι όμοιο με εκείνο του RAID3.

RAID 5 (1/2)

Το RAID5 αποτελεί μια τροποποίηση της δομής του RAID4 και προκύπτει από την παρατήρηση ότι ο δίσκος ισοτιμίας μπορεί μερικές φορές να είναι ο παράγοντας συμφόρησης του συστήματος (system bottleneck): στο RAID4, υπάρχει πρόσβαση στον δίσκο ισοτιμίας σε κάθε λειτουργία εγγραφής

Για να παρακάμψουμε αυτό το πρόβλημα, μπορούμε απλά να παρεμβάλλουμε τα μπλοκ ισοτιμίας ανάμεσα στους δίσκους. Με άλλα λόγια, δεν υπάρχει πλέον δίσκος μόνο για τα bits ισοτιμίας, αλλά ο κάθε δίσκος έχει κάποια μπλοκ δεδομένων και κάποια μπλοκ ισοτιμίας.

RAID 5 (2/2)



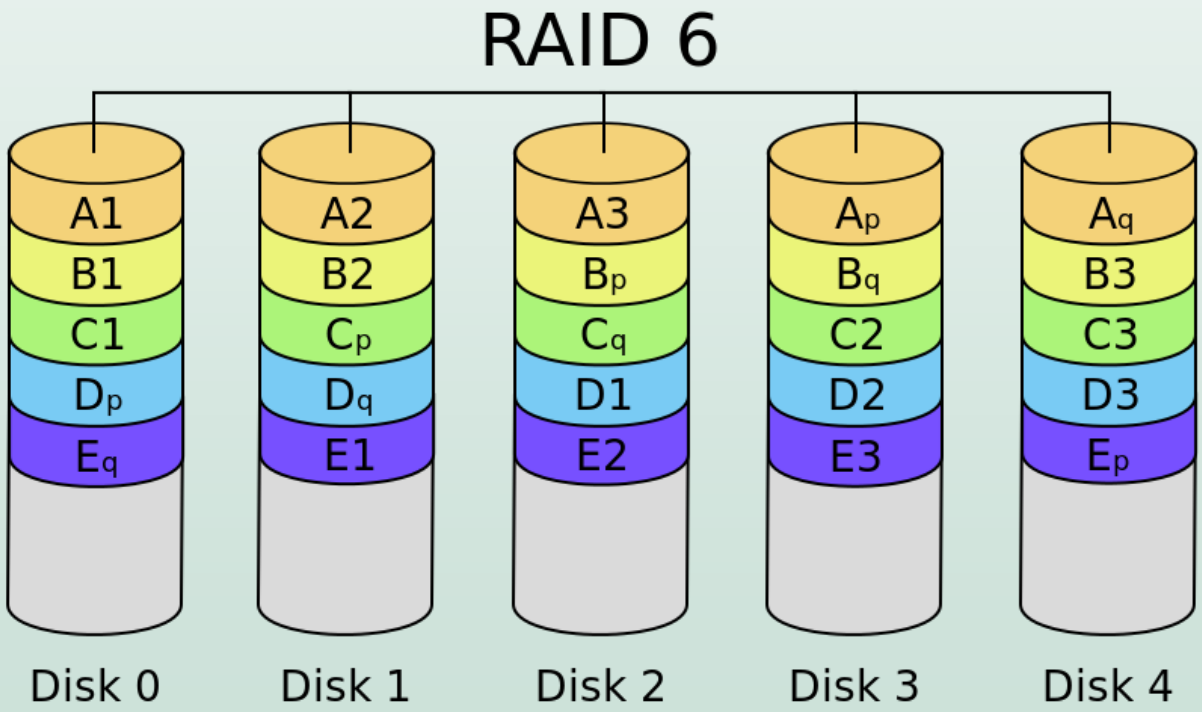
Κατανεμημένα μπλοκ ισοτιμίας σε σύστημα RAID5 πέντε δίσκων.

Το μοντέλο αξιοπιστίας για το RAID5 είναι προφανώς το ίδιο με το RAID4 και διαφέρουν μόνο στο μοντέλο απόδοσης.

RAID 6 (1/2)

Το RAID6 αποτελεί μια προέκταση του RAID5, προσθέτοντας ένα επιπλέον μπλοκ ισοτιμίας και κατανέμοντας τα δύο (πλέον) μπλοκ ισοτιμίας διαγώνια σε όλους τους δίσκους (όπως και το RAID5)

RAID 6 (2/2)


RAID 6
A1
B1
C1
Dp
Eq
A2
B2
Cp
Dq
E1
A3
Bp
Cq
D1
E2
Ap
Bq
C2
D2
E3
Aq
B3
C3
D3
Ep
Disk 0
Disk 1
Disk 2
Disk 3
Disk 4

Παράδειγμα - Εφαρμογή σχεδίασης συστήματος RAID (1/2)

Θέλουμε να εισάγουμε πληροφορία 1,2TB σε συστήματα δίσκων RAID0, RAID1, RAID4 και RAID5.
Όλοι οι δίσκοι που διαθέτουμε έχουν χωρητικότητα 400GB ο καθένας.

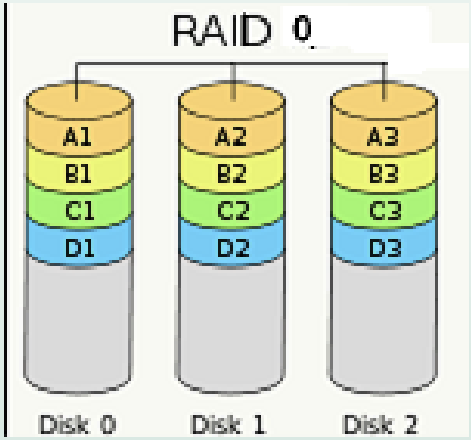
α. Πόσοι δίσκοι χρειάζονται σε κάθε περίπτωση;

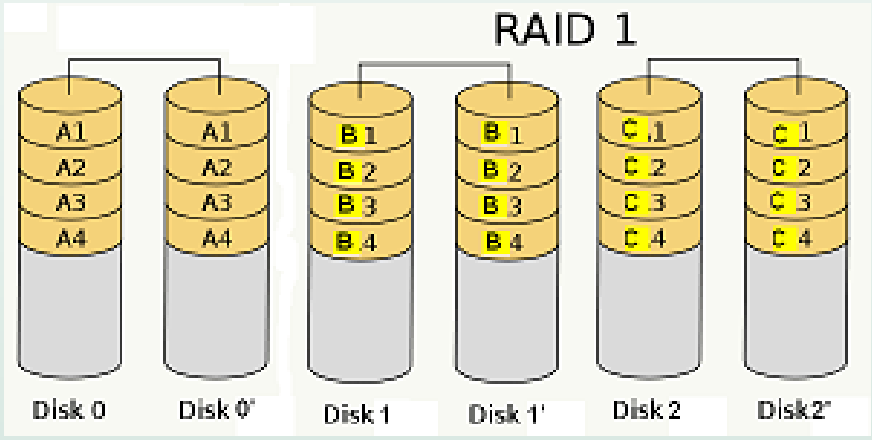
β. Να δείξετε (σχεδιάσετε) πως θα κατανεμηθεί η πληροφορία αυτή στους δίσκους (πληροφορίας και ισοτιμίας) σε κάθε περίπτωση.

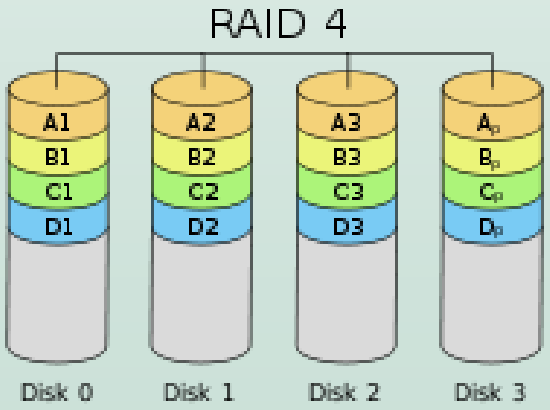
γ. Μέχρι πόσοι δίσκοι μπορούν να χάσουν ταυτόχρονα τα δεδομένα τους χωρίς να έχουμε απώλεια πληροφορίας;

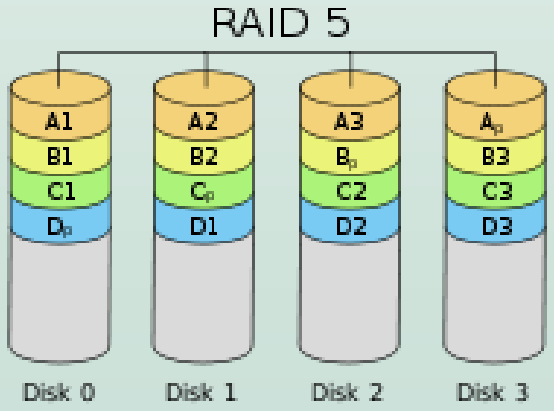
Υποθέστε πως το κάθε μπλοκ έχει μέγεθος 100GB (κάτι που δεν ισχύει στην πραγματικότητα, αλλά βολεύει για ευκολία πράξεων και σχεδιασμού).

Παράδειγμα - Εφαρμογή σχεδίασης συστήματος RAID (2/2)

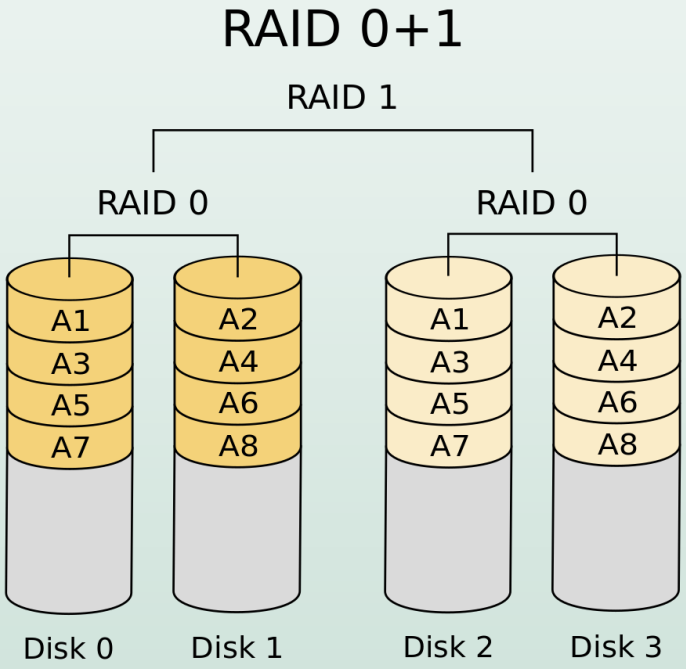


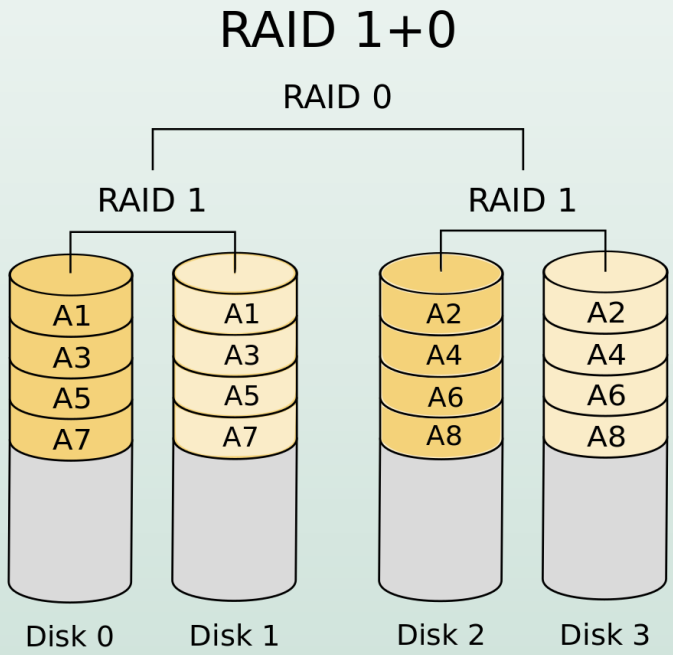






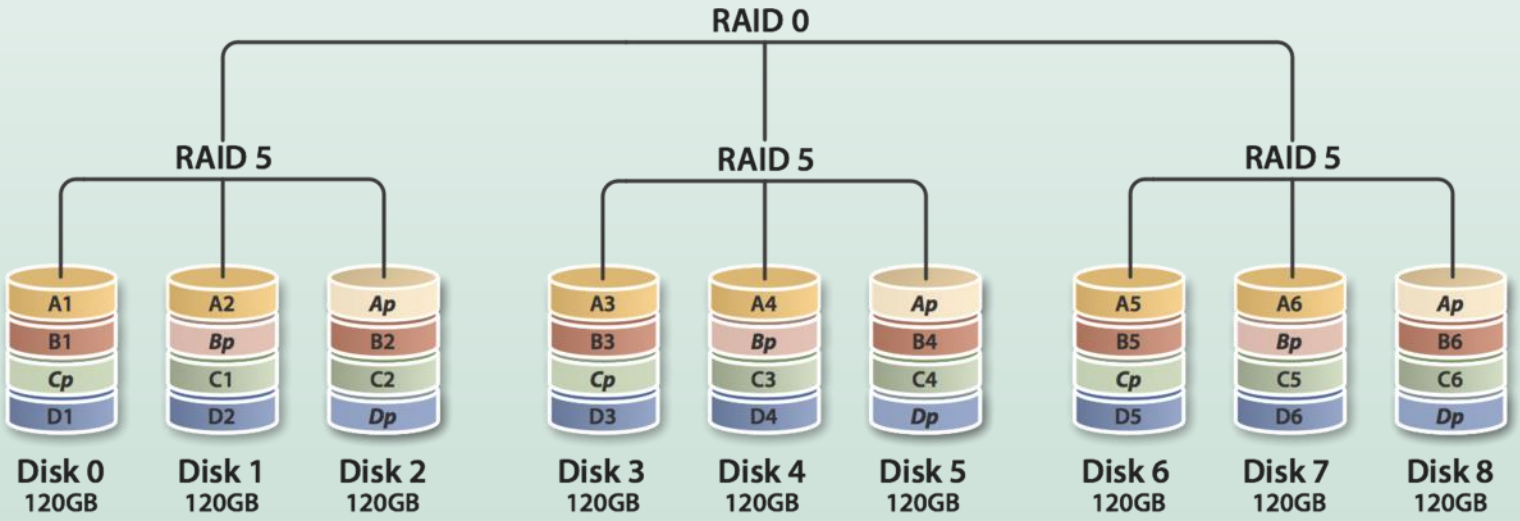
Nested RAID levels (1/3)





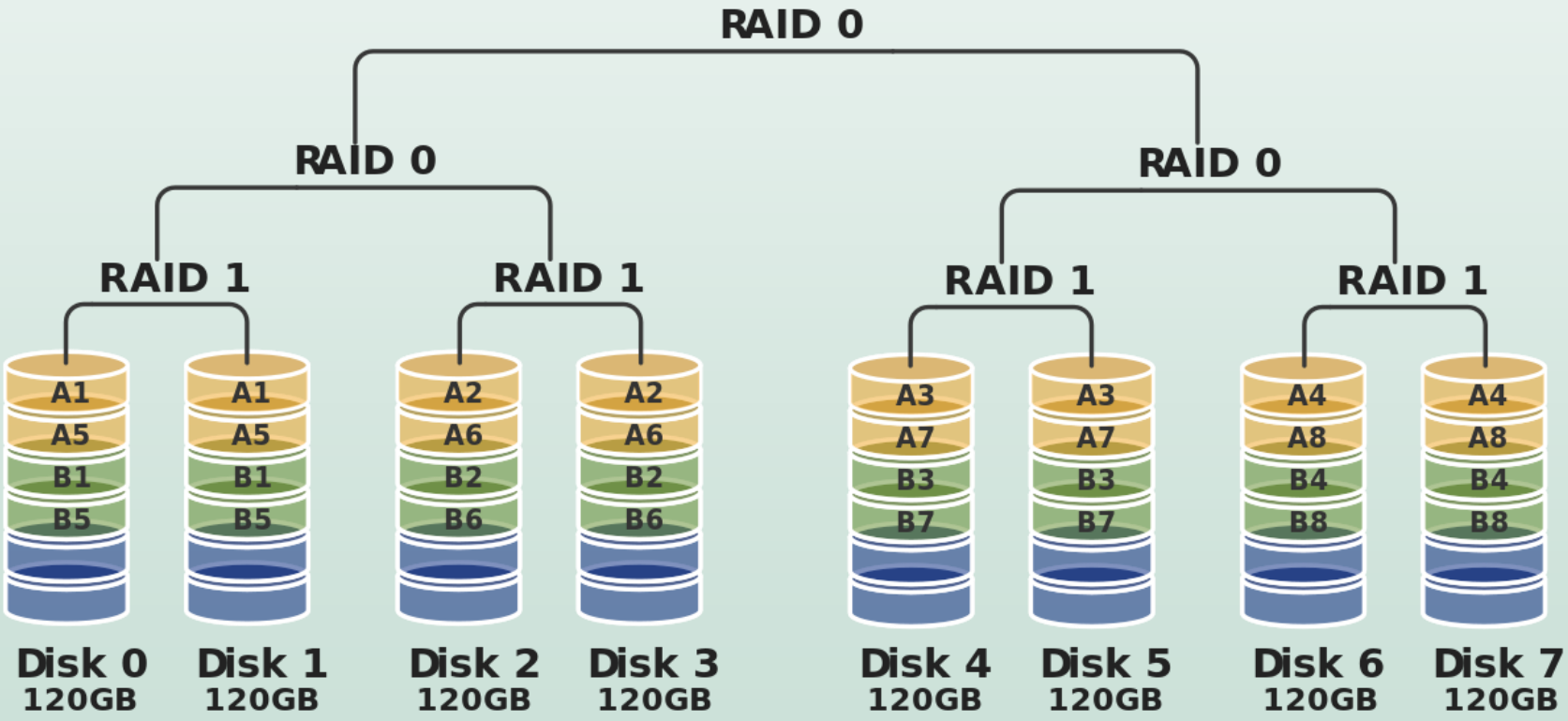
Nested RAID levels (2/3)

RAID 50

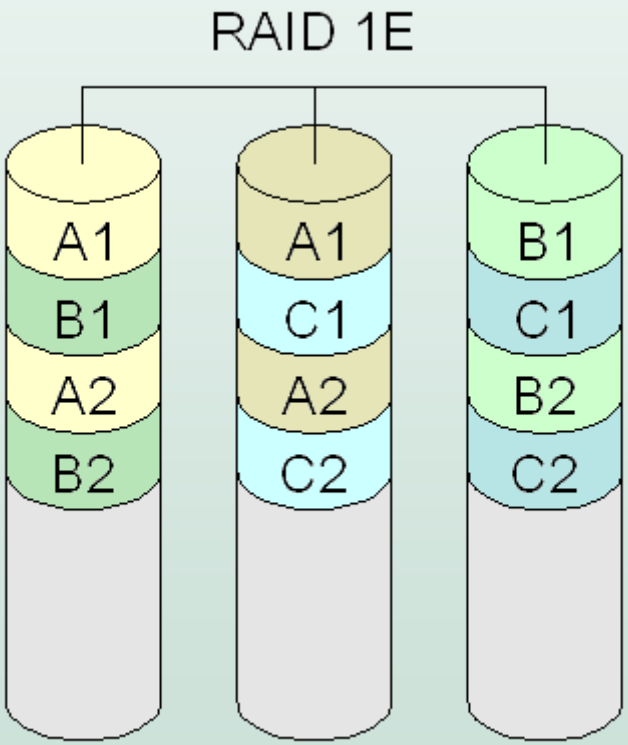


Nested RAID levels (3/3)

RAID 100



Non-standard RAID levels



Αναφορά - Υλικό

Fault-Tolerant,

Israel Koren and C. Mani Krishna,

Second Edition, 2020